

1 METHODS AND APPARATUS FOR IMPLEMENTING MULTIPLE TYPES OF NETWORK  
2 TUNNELING IN A UNIFORM MANNER

3  
4 BACKGROUND OF THE INVENTION

5  
6 1. Field of the Invention

7 The invention relates to datacommunications. More  
8 particularly, the invention relates to methods and apparatus for  
9 tunneling different types of data packets over different types of  
10 networks.

11  
12 2. State of the Art

13 Tunneling is a process whereby a data packet is encapsulated  
14 in another packet before traversing a network. There are two  
15 primary uses for tunneling. One use is to transport one type of  
16 packet over a network designed for another type of packet, e.g.  
17 Ethernet over ATM. Another application for tunneling is referred  
18 to as Virtual Private Networking, a process whereby a secure  
19 encrypted (or non-secure) connection is created across a public  
20 network through the use of tunneling.

21  
22 Currently there are a wide variety of tunneling protocols.  
23 Some are platform and/or network dependent. Among the most  
24 popular protocols are: IP (Internet Protocol) over IP, IP over

1 MPLS (multiprotocol label switching), Ethernet over MPLS, and L2TP  
2 (layer two tunneling protocol).

3

4 A Virtual Private Network (VPN) consists of two or more nodes  
5 connected by "virtual links", i.e. tunnels, through a public  
6 network such as the Internet. From the point of view of the  
7 nodes, the tunnel operates as a point to point link and the  
8 tunneling protocol operates as a link layer protocol.

9

10 By definition, a tunnel exists between two nodes. One node  
11 is referred to as the entry node and the other is referred to as  
12 the exit node. A tunnel is unidirectional. Bi-directional  
13 tunneling is achieved by pairing two tunnels. These are referred  
14 to as the "direct tunnel" and the "reverse tunnel". The two  
15 tunnels may traverse different nodes in the network or the tunnels  
16 may be symmetrical, i.e. traverse the same nodes in both  
17 directions.

18

19 Generally speaking, the tunneling process involves the  
20 processing of headers attached to data packets. For example, at  
21 the entry node one or more tunnel headers are pre-pended to the  
22 data packet. As the packet traverses the tunnel, intermediate  
23 nodes in the tunnel process the packet according to the tunneling  
24 protocol. For example, in IP over IP tunneling, intermediate  
25 nodes perform IP header processing and IP packet forwarding. In

1 an MPLS tunnel, intermediate nodes perform MPLS label processing  
2 and MPLS packet forwarding. At the exit node, destination node  
3 processing is performed. For example, in an IP over IP tunnel,  
4 the exit node reassembles IP packet fragments and processes the  
5 reassembled packets according to their original (inner) headers.  
6 The processing of headers or other packet information is performed  
7 with the aid of a database. The header information is used as a  
8 key to lookup a matching entry in the database to yield an output  
9 port. The following is a more detailed explanation of how IP,  
10 MPLS, and L2TP tunneling operate.

11

#### 12 IP over IP Tunneling

13

14 Each node in an IP tunnel maintains a "Forwarding Information  
15 Base" (FIB) which contains a plurality of entries. Each entry  
16 includes an IP address of a host or an IP prefix of one or more  
17 networks as well as information about the "Next Hop Routers"  
18 through which the destination host or network can be reached. (As  
19 used herein, the term router is meant to include switches as well  
20 as routers.) The Next Hop information consists of an IP address  
21 of the Next Hop router, the IP interface on which the Next Hop  
22 Router is reachable, and possibly more information such as a Layer  
23 2 address. The FIB entries are typically built based on routing  
24 information disseminated dynamically by IP Routing Protocols.

1 However, entries can also be built based on information statically  
2 configured into the router. Some routers may have multiple FIBs.

3  
4 IP packet forwarding is performed hop by hop. Each router in  
5 the tunnel uses the FIB information to find the best possible Next  
6 Hop router to forward an IP packet on its way to the final  
7 destination. The identity of the best possible Next Hop Router  
8 for forwarding an IP packet is determined by comparing the  
9 packet's destination IP address with the IP addresses and prefixes  
10 in the FIB. The FIB entry having the IP address or prefix which  
11 matches or most closely matches the destination IP address of the  
12 packet identifies the best possible Next Hop router. This  
13 comparison is referred to as the Longest Prefix Match. The  
14 Longest Prefix Match yields an output "IP interface" which is used  
15 to forward the packet.

16  
17 The "IP interface" (IF) is an abstraction of the IP functions  
18 on a physical or logical port of a router or switch. It leads to  
19 the identification of the egress port through the binding relation  
20 with the Layer 2 interface and performs the needed IP functions to  
21 pass the packet through the port. The IF is usually configured by  
22 the router operator with information which is relevant to the IP  
23 protocol processing performed by the router.

24

1        Each router used in IP over IP tunneling also maintains an  
2        "IP Tunnel Interface" (TIF) for each tunnel serviced by the  
3        router. The TIF is an abstraction of IP over IP tunnel functions  
4        on a physical or logical port of a router. It identifies the  
5        entry and exit nodes for the tunnel. Usually, it is configured by  
6        the router operator.

7  
8        Tunnels may be static or dynamic. A static tunnel uses the  
9        same set of routers and takes the same route through the network  
10       all the time. A dynamic tunnel can take different routes through  
11       the network based on network conditions or tunnel programming.  
12       Most IP tunnels are dynamic and use the best route available based  
13       on network congestion or time of day.

14  
15       When a packet reaches its final destination, the exit node  
16       router strips the tunnel header(s) from the packet. The remaining  
17       inner header is an IP header, and therefore, the packet is passed  
18       to an IP processing engine which performs an IP lookup on the  
19       inner IP header, i.e. the IP destination address. This lookup may  
20       yield an outgoing interface, if the packet is to be forwarded, or  
21       may indicate local consumption, if the router itself is the final  
22       destination. The router itself may be the final destination when  
23       the packet contains command/control information.

## 1 MPLS Tunneling

2  
3       The key concept in MPLS tunneling is identifying and marking  
4 packets with labels and forwarding them to a router which then  
5 uses the labels to forward the packets through the network. The  
6 labels are created and assigned to packets by a Label Distribution  
7 Protocol (LDP) based upon the information gathered from existing  
8 routing protocols or some other method.

9  
10       An MPLS tunnel includes a plurality of interconnected Label  
11 Switch Routers (LSRs). At least some of the LSRs are coupled to  
12 Label Edge Routers (LERs). An MPLS tunnel is also referred to as  
13 a Label Switched Path (LSP) from an input LER through LSRs to an  
14 output LER. When a packet arrives at an LER, the LER extracts the  
15 datagram (the data portion of a packet) and the routing  
16 information from the packet and assigns a label to the datagram  
17 based on routing information. The datagram with the label is then  
18 sent to an LSR based on the label. The LSR which receives the  
19 datagram forwards it on through the network based on the label.

20  
21       An LSP is a set of LSRs that packets belonging to a certain  
22 FEC (forwarding equivalence class) travel in order to reach their  
23 destination. Each LER in an MPLS tunnel (LSP) has an Incoming  
24 Label Map (ILM). The ILM specifies the action to take when a  
25 labeled packet is received. Each entry defines an incoming label,

1 a label operation, and a link to a Next Hop Label Forwarding Entry  
2 (NHLFE). The ILM is built based on label distribution information  
3 disseminated by a Label Distribution Protocol (LDP) engine.

4  
5 Each LER in the MPLS tunnel (LSP) has an FEC-to-NHLFE Map  
6 (FTN). The FTN specifies the action to take when an unlabeled  
7 packet is received. Each entry in the FTN defines a set of  
8 characteristics used to categorize the packet, and a link to an  
9 NHLFE. The FTN is also built based on information disseminated by  
10 a Label Distribution Protocol (LDP) engine.

11  
12 An NHLFE specifies "how to" forward a packet. It defines an  
13 outgoing label, a label operation, a next hop IP address, and an  
14 output interface. It may also specify an MPLS label stack. The  
15 label stack identifies a series of labels to push on a labeled  
16 packet in the process of forwarding the packet on an LSP segment.  
17 The NHLFE is built based on label distribution information  
18 disseminated by a Label Distribution Protocol (LDP) engine.

19  
20 Simple label forwarding is realized by pointing an ILM entry  
21 to an NHLFE, and applying a label swap. This is referred to as  
22 the MPLS label swapping/forwarding function. An LSP is originated  
23 by pointing an FTN entry to a NHLFE, and applying a label push.  
24 This is referred to as the LSP entry function. An LSP is  
25 terminated by not pointing an ILM entry to any NHLFE, and applying

1 a label pop at input. Terminating the LSP does not guarantee that  
2 the packet will be consumed by the local node. The packet is  
3 forwarded using the exposed label or IP header. The collection of  
4 ILMs and related NHLFEs are collectively called the MPLS Label  
5 Information Base or LIB.

## 6 7 L2TP Tunneling

8  
9 The L2TP (layer two tunneling protocol) is an extension to  
10 PPP (point-to-point protocol) that enables ISPs to operate Virtual  
11 Private Networks (VPNs). L2TP works with UDP (user datagram  
12 protocol) and IP drivers. L2TP uses a "tunnel list" which is  
13 analogous to an MPLS LIB. When a packet is received from a PPP  
14 link, part of the PPP header is stripped off and replaced with an  
15 L2TP header which includes tunnel and session IDs. The packet is  
16 sent to an L2TP tunnel by writing to a UDP driver with IP  
17 interface data and the end point IP address. At the end of the  
18 tunnel, data is read from a UDP port. The tunnel ID, session ID,  
19 and packet flags are extracted. The data is then formatted and  
20 written to a PPP link associated with the session. L2TP provides  
21 a "one hop" virtual PPP link which spans a multi-hop IP path.

22  
23 From the foregoing it will be appreciated that the different  
24 tunneling protocols process packets in very different ways. State  
25 of the art routers which are intended to support different types



1 of tunneling have separate processing engines for each supported  
2 tunneling protocol. Depending on the number of tunneling  
3 protocols to be implemented, the router may not be cost effective  
4 from the perspective of resources, design time and  
5 maintainability.

#### 6 7 SUMMARY OF THE INVENTION

8  
9 It is therefore an object of the invention to implement  
10 multiple tunneling protocols in a switch or router.

11  
12 It is also an object of the invention to implement multiple  
13 tunneling protocols in a switch or router in a cost effective way.

14  
15 It is another object of the invention to implement multiple  
16 tunneling protocols in a switch or router using the fewest  
17 possible resources.

18  
19 It is still another object of the invention to implement  
20 multiple tunneling protocols in a switch or router with efficient  
21 design time.

22  
23 It is also an object of the invention to implement multiple  
24 tunneling protocols in a switch or router which is easy to  
25 maintain.

1 Another object of the invention is to implement multiple  
2 tunneling protocols in a switch or router using a single general  
3 processing engine.  
4

5 In accord with these objects which will be discussed in  
6 detail below, the present invention provides a uniform method for  
7 implementing multiple tunneling protocols in a switch or router.  
8 The invention is based on the realization that although the  
9 tunneling protocols are very different, they do share a similar  
10 overall structure which can be exploited to create a unified  
11 method of dealing with multiple protocols. By using similar data  
12 structures to implement multiple protocols, the invention makes  
13 data management and programming simple and, therefore, cost  
14 effective.  
15

16 According to the invention, all tunneling protocols are  
17 abstracted as the mapping of input L2 or L3 streams with output L2  
18 or L3 streams. An L2 or L3 interface is an abstraction of a  
19 physical or logical port in a router. According to the invention,  
20 mapping is provided by a finite set of tunnel interfaces. Each  
21 tunnel interface is a logical entity that is characterized by a  
22 set of tunnel specific attributes; these attributes include, for  
23 example, the parameters identifying tunnel end points. At the  
24 tunnel origination or termination point in the network, incoming  
25 streams arriving on an input port are mapped to tunnel interfaces.

1 The tunnel interfaces, in turn, map the streams to output  
2 interfaces. As traffic streams flow through these interfaces,  
3 they are processed according to defined attributes of these  
4 interfaces. The interface attributes are tunnel end-point  
5 specific (i.e., start or end of a tunnel). Mapping is performed  
6 by using context data in an arriving packet as a search key to a  
7 database.

8  
9 At tunnel origination, e.g., the tunnel entry-point node, for  
10 all types of tunnels considered by this invention, a first  
11 database lookup identifies a tunnel interface appropriate for the  
12 packet processing. Once the tunnel interface is selected, the  
13 processing continues according to information associated with the  
14 tunnel interface. In the case where the tunnel is layered over  
15 IP, there are two options. In the first option, the output  
16 interface information is cached. Caching the output interface  
17 information requires a refresh, in case forwarding information  
18 base updating results in changing the path to a different next hop  
19 router. In the second option, a second database lookup is  
20 employed, on an FIB associated with the tunnel interface, to find  
21 the best choice for the outgoing IP interface. In cases where  
22 applicable and necessary, IP segmentation and re-assembly of  
23 packets is performed to meet the MTU (maximum transmission unit)  
24 requirements of the interface.

25

1       At tunnel termination, e.g., the tunnel exit-point node, for  
2 all types of tunnels considered by this invention, a first  
3 database lookup identifies the end of the tunnel and an interface  
4 associated with the type of processing of the inner header  
5 remaining after tunnel header decapsulation. The inner header of  
6 the packet, and a database associated with the interface  
7 identified by the first database lookup may be used to perform a  
8 second lookup, yielding the outgoing interface for the packet.

9  
10       The methods of the invention provide similar structuring of  
11 processing engines for all supported tunneling protocols. For  
12 example, for each supported protocol, the invention provides an  
13 input interface, an output interface, an information base, a  
14 mapping tunnel interface and a mapping information base.

15  
16       The invention also provides an API for programming the host  
17 processor of a router or switch to perform the methods of the  
18 invention.

19  
20       Additional objects and advantages of the invention will  
21 become apparent to those skilled in the art upon reference to the  
22 detailed description taken in conjunction with the provided  
23 figures.

24  
25                   BRIEF DESCRIPTION OF THE DRAWINGS

1        Figure 1 is a high level schematic diagram illustrating the  
2 mapping of input, output and tunnel interfaces according to the  
3 invention;

4  
5        Figure 2 is a high level schematic diagram illustrating  
6 tunnel origination processing according to the invention;

7  
8        Figure 3 is a high level schematic diagram illustrating  
9 tunnel termination processing according to the invention;

10  
11       Figure 4 is a table comparing the interfaces and information  
12 databases for the transmit side of four types of tunnels; and

13  
14       Figure 5 is a table comparing the interfaces and information  
15 databases for the receive side of four types of tunnels.

16  
17                    BRIEF DESCRIPTION OF THE APPENDIX

18  
19       The attached CDROM appendix includes a source code  
20 description of an API useful for implementing the methods of the  
21 invention.

## 1 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

2  
3 Referring now to Figure 1, input and output streams are  
4 mapped to tunnel interfaces as relational sets. More  
5 particularly, input L2 streams  $iL21-iL2n$  are treated separately  
6 from input L3 streams  $iL31-iL3n$ . Separate sets of tunnel  
7 interfaces  $T1-Tm$  are provided for the L2 and L3 traffic. Output L2  
8 streams  $eL21-eL2n$  are treated separately from output L3 streams  
9  $eL31-eL3n$ . However, as seen in Figure 1, streams that enter the  
10 router as L2 may exit as L3 streams and vice versa. As shown in  
11 Figure 1, input L2 streams are mapped to tunnel interfaces  $T$  by  
12 forwarding function  $f2i$  and input L3 streams are mapped to tunnel  
13 interfaces  $T$  by forwarding function  $f3i$ . L2 tunnel interfaces are  
14 mapped to L2 output interfaces  $eL21-eL2n$  by forwarding function  
15  $f2e$  and L3 tunnel interfaces are mapped to L3 output streams by  
16 forwarding function  $f3e$ . Forwarding function  $f23e$  maps input L2  
17 streams from their tunnel interface to an output L3 stream and  
18 forwarding function  $f32e$  maps input L3 streams from their tunnel  
19 interface to an output L2 interface.

20  
21 The relationship of the streams and interfaces shown in  
22 Figure 1 can be described as a relationship between sets as  
23 defined by the forwarding functions. For example:

24  
25  $F2i$ :  $iL2$  maps to  $T$ , given  $iL2$  and  $T$  don't belong to null sets;

1 F2e: T maps to eL2, given T and eL2 don't belong to null sets;  
2 F23e: T maps to eL3, given T and eL3 don't belong to null sets;  
3 F3i: iL3 maps to T, given iL3 and T don't belong to null sets;  
4 F3e: T maps to eL3, given T and eL3 don't belong to null sets; and  
5 F32e: T maps to eL2; given T and eL2 don't belong to null sets.  
6

7 The mapping (forwarding function) is performed with the aid  
8 of context data that a packet (unit of traffic in a stream)  
9 carries or is associated with and database information which is  
10 configured and updated by a local host.  
11

12 Turning now to Figure 2, the tunnel origination processing  
13 according to the invention is illustrated with respect to input  
14 and output L2 and L3 interfaces, tunnel interfaces, associated  
15 databases, and forwarding functions (data base lookups). As an IP  
16 packet arrives from an IP interface 10, its destination address is  
17 retrieved and used as search key by the forwarding function IP  
18 forwarding Lookup 110 to find the longest prefix match in the  
19 forwarding information base FIB 210. The forwarding information  
20 base lookup yields a particular tunnel interface, e.g. IP in IP  
21 tunnel interface 20 or MPLS tunnel interface 22. The tunnel  
22 interface points to either an L3 interface 30 or an L2 interface  
23 32.  
24

1        In the case of IP in IP tunneling, an IP header (the tunnel  
2 header) is constructed based on information held in the tunnel  
3 interface 20. The IP header is prepended to the packet by the  
4 encapsulation engine 21. If the IP output information is cached,  
5 the tunnel interface 20 will provide the information pointing to  
6 output interface 30, and the packet will be forwarded directly to  
7 that interface. If output interface caching is not used, the  
8 tunnel interface 20 provides an association with an FIB. This FIB  
9 can be an FIB 220 specific to this interface, or it can be the FIB  
10 210 used by all IP interfaces 10. The L3 processing employs an IP  
11 lookup mechanism 110 to search this FIB. The IP header information  
12 from the tunnel header is used in this search. This lookup yields  
13 the output interface 30 to which the packet is forwarded.

14  
15        In the case of IP over MPLS, an MPLS header (tunnel header)  
16 is prepended to the packet based on information held in the tunnel  
17 interface 22. The tunnel interface 22 provides the Output Label,  
18 or a stack of Output Labels, that are stored in this header by the  
19 encapsulation engine 23. The resulting packets are then forwarded  
20 to the L2 Interface 32.

21  
22        When packets arrive at PPP interface 12, the PPP context  
23 information is used as a search key by the PPP forwarding lookup  
24 112 into the TSIB (tunnel session information base) 212. This



1 yields an L2TP tunnel interface 24 which points to an L3 output  
2 interface 34.

3  
4 In the case of L2TP tunneling, a set of headers, collectively  
5 called the L2TP header, is created based on information from the  
6 L2TP tunnel interface 24 and are prepended to the packet by the  
7 encapsulation engine 25. If the L3 (IP) or L2 output interface  
8 information is cached, the L2TP tunnel interface 24 will provide  
9 the information pointing to that output interface 34. If output  
10 interface caching is not used, the L2TP tunnel interface 24  
11 provides an association to the FIB 222, which the L3 forwarding  
12 engine IP lookup 110 will search. The L2TP interface may have its  
13 own FIB 222 or there may be only one FIB 210 in the entire system.

14  
15 When Ethernet packets arrive at the Ethernet interface 14,  
16 the Ethernet MAC and/or VLAN tag is retrieved and used as search  
17 key by the Ethernet forwarding lookup 114 to search the switching  
18 information base SIB 214. This points to an MPLS tunnel interface  
19 26 which points to an L2 interface 36. The MPLS tunnel interface  
20 26 provides the Output Label or a Stack of Labels, which are  
21 filled out in the MPLS header prepended to the packet by the  
22 encapsulation engine 27. The resulting packets are forwarded to  
23 the output L2 Interface 36.

1       As shown in Figure 2, the tunnel interfaces are marked "IP in  
2 IP Tunnel Interface", "MPLS Tunnel Interface" (for IP), "L2TP  
3 Tunnel Interface", and "MPLS Tunnel Interface" (for Ethernet).  
4 These are just the "type" of interfaces, but for each type, there  
5 can be many interfaces, for example many "IP in IP", or many  
6 "MPLS" interfaces.

7  
8       Figure 3 illustrates tunnel termination processing according  
9 to the invention.

10  
11       The input interface 50 is the terminus of all IP tunnels and  
12 L2TP tunnels. The IP forwarding lookup 110 uses the received  
13 packet header information to perform an IP lookup of the FIB 250  
14 which yields the tunnel termination interface 60 or 62. If the  
15 packet exits the tunnel through the IP in IP tunnel interface 60,  
16 the IP tunnel header is dropped, exposing the inner IP header.  
17 The inner IP header is processed by the decapsulation engine 61.  
18 The Tunnel interface 60 is associated with its own FIB 260, or a  
19 single system FIB 210. A second IP lookup 110 is performed on  
20 this FIB, which yields an L3, or L2 output interface 70 on which  
21 the packet is forwarded

22  
23       In the case of L2TP tunnels, the IP forwarding lookup 110  
24 searches the FIB 250 (if a separate FIB is provided or FIB 210 if  
25 a single FIB is shared with other lookup functions), which yields

1 the L2TP tunnel termination interface 62, and an indication of the  
2 L2TP tunnel termination processing type. The IP header and UDP  
3 header are dropped by the decapsulation engine, exposing the L2TP  
4 header. Tunnel ID and Session ID information from this header is  
5 used to perform a second lookup 162, on the L2TP information data  
6 base 262 which yields the PPP output interface 72 to which the  
7 packet is forwarded.

8  
9 The interface 52 is the terminus of all MPLS tunnels. The  
10 decapsulation engine 53 uses the incoming packet's label  
11 information to perform a first (MPLS) lookup 152 on the LIB 252  
12 associated with the MPLS input interface 52. This yields one of  
13 the following:

14 a virtual connection (VC) ID, and its attached Ethernet  
15 interface 62,

16 a virtual LAN ID, or a destination MAC address, and the  
17 attached Ethernet interface 62, or

18 an IP interface 66.

19  
20 For Ethernet over MPLS, after popping the MPLS label, the  
21 inner Ethernet header is exposed. The Ethernet interface 62 has an  
22 associated data base SIB 262 (or 214, if there is only one SIB in  
23 the system). The information from the Ethernet header is used for  
24 a second, Ethernet, lookup which is performed on this data base

1 262 or 214. This yields the output interface 74 to which the  
2 packet is forwarded.

3  
4 For IP over MPLS tunnels, after popping the MPLS label, the  
5 inner IP header is exposed. The information in the IP header is  
6 used for a second, IP lookup 110. This is performed on the FIB  
7 266 associated with the IP interface 66 (or 210 if there is only  
8 one FIB in the system) which points to L3 or L2 output interface  
9 76.

10  
11 From the foregoing, it will be appreciated that the methods  
12 described thus far enable the implementation of four different  
13 kinds of tunnels using similar data structures for each. Figure 4  
14 and Figure 5 illustrate the corresponding data structures.

15  
16 As shown in Figure 4, for Tunnel Origination (Tunnel Entry)  
17 processing, the invention provides for each protocol an "input  
18 interface", an "input information database", a "mapping transmit  
19 interface", a "mapping information database", and an "output  
20 interface". The "mapping transmit interface" is the transmit side  
21 of the tunnel interface.

22  
23 In the case of IP over IP tunnel origination, the input  
24 interface is an IP interface, the input information database is an  
25 FIB, the mapping transmit interface is an IP in IP transmit tunnel

1 interface, the mapping information database is optional and may be  
2 an FIB, and the output interface is an IP or L2 interface.

3  
4 In the case of L2TP tunnel origination, the input interface  
5 is a PPP interface, the input information database is a TSIB, the  
6 mapping transmit interface is an L2TP transmit tunnel interface,  
7 the mapping information database is optional and may be an FIB,  
8 and the output interface is an IP or L2 interface.

9  
10 In the case of IP over MPLS tunnel origination, the input  
11 interface is an IP interface, the input information database is an  
12 FIB, the mapping transmit interface is an MPLS transmit tunnel  
13 interface, there is no mapping information database, and the  
14 output interface is an L2 interface.

15  
16 In the case of Ethernet over MPLS tunnel origination, the  
17 input interface is an Ethernet interface, the input information  
18 database is an SIB, the mapping transmit interface is an MPLS  
19 transmit tunnel interface, there is no mapping information  
20 database, and the output interface is an L2 interface.

21  
22 As shown in Figure 5, for Tunnel Termination (Tunnel Exit)  
23 processing, the invention provides for each protocol an "input  
24 interface", an "input information database", a "mapping receive  
25 interface", a "mapping information database" and an "output

1 interface". The "mapping receive interface" is the receive side of  
2 the tunnel interface.

3  
4 In the case of IP over IP tunnel termination, the input  
5 interface is an IP interface, the input information database is an  
6 FIB, the mapping receive interface is an IP in IP receive tunnel  
7 interface, the mapping information database is an FIB, and the  
8 output interface is an IP or L2 interface.

9  
10 In the case of L2TP tunnel termination, the input interface  
11 is an IP interface, the input information database is an FIB, the  
12 mapping receive interface is an L2TP receive tunnel interface, the  
13 mapping information database is a TSIB, and the output interface  
14 is a PPP interface.

15  
16 In the case of IP over MPLS tunnel termination, the input  
17 interface is an MPLS interface, the input information database is  
18 an LIB, the mapping receive interface is an IP interface, the  
19 mapping information database is an FIB, and the output interface  
20 is an IP or L2 interface.

21  
22 In the case of Ethernet over MPLS tunnel termination, the  
23 input interface is an MPLS interface, the input information  
24 database is an LIB, the mapping receive interface is an Ethernet

1 interface, the mapping information database is an SIB, and the  
2 output interface is an Ethernet interface.

3  
4 According to the present implementation of the invention,  
5 tunnel interfaces are of two types. One type is referred to as a  
6 "transmit" interface, which is associated with entering the  
7 tunnel. The other is referred to as a "receive" tunnel interface,  
8 which is associated with exiting the tunnel. Each "end" of a  
9 tunnel will use at least a transmit tunnel interface (entry in  
10 tunnel), or a receive tunnel interface (exit from tunnel).  
11 Bidirectional tunnels will have both types of interface at each  
12 end.

13  
14 The transmit tunnel interface is characterized by the  
15 following parameters: tunneling protocol, encapsulation header  
16 field values, such as local source address, remote destination  
17 address, hop limit, and tunnel MTU (for IP in IP, or L2TP), or  
18 MPLS output label(s), L2TP header fields.

19  
20 According to the illustrated embodiment, four protocols are  
21 supported: IP in IP (where IP can be either IPv4 or IPv6), IP over  
22 MPLS, Ethernet over MPLS, and L2TP. For IP in IP (IPv4 and IPv6)  
23 and L2TP the local source address is the address of tunnel entry  
24 node, the remote destination address is the address of the tunnel  
25 exit node, and the hop limit is the number of hops or "time to

1 live" set in the tunnel header. The tunnel MTU is the parent  
2 interface MTU less the tunnel header size.

3  
4 The "receive" tunnel interface requires fewer parameters.  
5 The invention provides similar structuring of Tunnel Interfaces  
6 (receive interfaces and transmit interfaces) and similar logical  
7 linking between data structures used for input packet processing.  
8 The receive interface yields the structure used for IP lookup  
9 (FIB) or MPLS label mapping (LIB), or L2TP Tunnel and Session ID  
10 mapping (TSIB)), and similar logical linking between data  
11 structures used for output packet processing. The transmit  
12 interface provides the information about the encapsulation of the  
13 packet, tunnel IP header source and destination addresses, or MPLS  
14 Labels, or L2TP IP source and destination addresses.

15  
16 Turning now to the Appendix, the tunnel transmit interface is  
17 characterized by the following basic parameters:

18 Tunneling protocol - the tunneling protocol can be IPv4 in  
19 IPv4, IPv6 in IPv6, GRE, etc.,...

20 Local source address - address of tunnel-entry node

21 Remote destination address - address of tunnel-exit node

22 Hop limit - the number of hops or time to live set in the  
23 tunnel header

24 Tunnel MTU - the parent interface MTU less the tunnel header  
25 size.



1       IPv4 tunnel interface attributes are illustrated at lines 82-  
2 95 of the Appendix and IPv6 tunnel interface attributes are  
3 illustrated at lines 96-113 of the Appendix. IPv4 source and  
4 destination address definitions are illustrated at lines 159-166  
5 and IPv6 source and destination address definitions are  
6 illustrated at lines 167-174 of the Appendix. The function to set  
7 IP source and destination address is illustrated at lines 182-190  
8 of the Appendix.

9  
10       A tunnel interface for MPLS tunneling can be created only if  
11 a layer 2 interface exists, with at least one child layer 3  
12 interface. For MPLS tunnels, the transmit interface holds the  
13 MPLS encapsulation information - label stack, and actions to be  
14 performed. The MPLS receive interface is associated with  
15 an LIB. It receives MPLS packets and helps locate the LIB used  
16 for Input Label Match. An MPLS tunnel interface is a logical  
17 interface on which MPLS packets are received and transmitted.  
18 MPLS tunnel interface attributes are illustrated at lines 114-128.  
19 A function to set MPLS tunnel label stacks on a set of interfaces  
20 is illustrated at lines 192-201.

21  
22       L2TP tunnel interface attributes are illustrated at lines  
23 133-148 of the Appendix. A function to set L2TP tunnel interface  
24 attributes is illustrated at lines 225-237.

1       A function to associate an information base with a tunnel  
2 interface is illustrated at lines 211-222 of the Appendix.

3  
4       Error codes are illustrated at lines 239-294 of the Appendix.

5  
6       There have been described and illustrated herein a uniform  
7 method for implementing multiple tunneling protocols. While  
8 particular embodiments of the invention have been described, it is  
9 not intended that the invention be limited thereto, as it is  
10 intended that the invention be as broad in scope as the art will  
11 allow and that the specification be read likewise. It will  
12 therefore be appreciated by those skilled in the art that yet  
13 other modifications could be made to the provided invention  
14 without deviating from its spirit and scope as so claimed.